САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Григорян Давид Артурович

Выпускная квалификационная работа бакалавра

Алгоритм обоснования операции на основе анализа данных группы пациентов

Направление 010400

Прикладная математика и информатика

Научный руководитель,
кандидат физ.-мат. наук,
доцент
Добрынин В. Ю.

Санкт-Петербург

2017

Содержание

Введение

На данный момент, IT — технологии играют большую роль в жизни человека. В последнее время все актуальнее становится вопрос внедрения машинного обучения в различные сферы человеческой деятельности. Одними из основных задач машинного обучения являются задачи классификации, кластеризации и восстановления регрессии.

Оно применяется в распознавании речи и текста (в частности, рукописного ввода), распознавании образов на изображениях и в видеопотоке, обнаружении спама, информационном поиске и т. д. Впоследствии, результаты работы искусственного интеллекта используются при создании автопилотов, голосовых ассистентов, рекомендательных систем, интернета вещей.

Также одной из областей применения машинного обучения является медицинская диагностика. Например, в 2015 году сотрудниками Массачусетского технологического университета была разработана система, позволяющая диагностировать у пациента болезнь Альцгеймера или болезнь Паркинсона по оцифрованному изображению стрелочных часов, нарисованному пациентом (dCDT - digital Clock Drawing Test). Точность работы данной системы была достаточно высока и достигала значений от 75 до 93 процентов в зависимости от используемого алгоритма [1].

Впрочем, популярность набирает не только машинное обучение. Все больше предприятий и компаний используют ERP – системы (Enterprise Resource Planning). Это программное обеспечение представляет из себя интегрированный набор бизнес – приложений. Они позволяют использовать единую модель данных, хранящую в себе информацию о финансах, управления персоналом, обслуживании, поставках и прочем. Также, ERP – системы позволяют автоматизировать бизнес-процессы в различных отраслях производства, что положительно сказывается на их эффективности [2].

ERP – системы для учреждений здравоохранения имеют специальное

название – медицинские информационные системы (МИС). Они позволяют вести электронные медицинские карты, автоматизировать деятельность лечебно-профилактических учреждений [3]. Работать с данными одной из таких МИС нам и придется.

Постановка задачи

Нам дан дамп базы данных SQL МИС «Виста-Мед», используемой в сердечно-сосудистом отделении одной из больниц. Перед нами стоит несколько крупных задач.

Во-первых, необходимо создать классификатор, который на основе некоторых признаков будет ставить диагноз о необходимости операции тому или иному пациенту. Очевидно, что система будет носить рекомендательный характер и окончательное решение принимает лечащий врач.

Нужно будет выбрать признаки, на основе которых будет производиться классификация. Желательно выбрать несколько разных наборов признаков и посмотреть, какая точность работы у системы в зависимости от выбранных признаков.

Возможно, размер полученной выборки будет мал. Тогда, нам потребуется применить кросс-валидацию (перекрестную проверку), чтобы узнать объективную точность работы модели.

Во-вторых, планируется обобщить задачу диагностики операции до задачи классификации пациентов по болезням. Неправильно поставленный диагноз может плачевно сказаться на здоровье пациента. Поэтому, поставленная задача имеет высокую практическую ценность.

В-третьих, велика вероятность того, что полученные выборки будут несбалансированными. Так, например, в статье «A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data» [4] (Gustavo E. A. P. A. Batista, Ronaldo C. Prati, Maria Carolina Monard) упоминается, что несбалансированные данные могут встречаться в базах данных медицинских записей. Если такое произойдет, придется использовать методы, позволяющие сбалансировать данные в выборке.

Обзор литературы

В качестве примера использования технологий машинного обучения в медицине рассматривается работа [1]. В ней, ученые из Массачусетского технологического университета рассматривают задачу постановки диагноза об болезни Альцгеймера или Паркинсона, используя стандартный в медицине тест рисования часов. Из каждого изображения извлекаются некоторые числовые характеристики, и с помощью различных алгоритмов построения дерева решений, метода опорных векторов, случайного леса и других методов строится классификатор. Авторам удалось достичь достаточно высоких результатов (см. Введение).

Для обзора ERP-систем и, в частности, МИС использовались материалы [2,3]. В первом источнике, дается подробное определение ERP – системы, а во втором выполнен обзор рынка медицинских информационных систем за 2005-2011 гг.

Книга [7] содержит сведения об МКБ – международном классификаторе болезней, информацию об его структуре и устройстве, а также расшифровки всех индексов, которыми обозначается каждое конкретное заболевание. Она была нужна, чтобы правильно выбрать необходимый нам класс болезней.

Очень полезным оказался учебник К. В. Воронцова «Математические методы обучения по прецедентам (теория обучения машин)» [6]. В нем разобраны большое количество методов машинного обучения, грамотно поставлены задача классификации с точки зрения математики.

Также использовались источники, связанные с медициной: [8] для обоснования выбора основного клинического анализа крови в качестве признака для классификации и [20] для того чтобы узнать, какую информацию несет в себе биохимический анализ крови.

В работе использовались следующие методы классификации: метод опорных векторов (SVM), дерево решений, случайный лес (random forest). В конспекте лекций К. В. Воронцова по методу опорных векторов [11] коротко

и ясно описаны основные аспекты этого метода, случаи линейной разделимости и неразделимости. Материалы [12-15] содержат справочную информацию об деревьях решений, методах построения и критериях, используемых при построении. В публикации [16] описывается алгоритм случайного леса, рассматриваются результаты, полученные при его применении на различных выборках и сравниваются с результатами, полученными при использовании алгоритмов AdaBoost и дерева решений.

Для введения метрик качества классификатора использовалась книга Маннинга К., Рагхавана П., Шютце Х. «Введение в информационный поиск» [17]. В ней, в главе «Оценка информационного поиска» описаны как находятся оценки классификатора – точность (precision), полнота (recall), f-мера, правильность (accuracy), что они означают и почему правильность не всегда подходит для оценки классификатора.

В работе [18] рассматриваются такие методы оценки классификатора, как кросс-валидация (cross-validation) и бутстрап (bootstrap). Оценивается влияние различных параметров на эти методы при использовании реальных наборов данных. Результаты, полученные авторами, показывают, что наилучшие результаты достигаются при разделении выборки на 10 частей, даже если вычислительные мощности позволяют разделить ее на большее количество частей.

Ресурс [21] содержит информацию о том, как бороться с проблемой несбалансированных выборок. В частности, предлагается попробовать собрать больше данных, или изменить размер выборки, используя oversampling или undersampling.

В публикации [22] рассматривается алгоритм SMOTE. Он позволяет получить высокую производительность модели машинного обучения на несбалансированных выборках путем генерации дополнительных объектов класса, в котором мало элементов. Такой подход называется oversampling. Авторы показывают, что результаты при использовании алгоритма лучше, чем при обучении на необработанной несбалансированной выборке.

Работа [23] описывает алгоритм ADASYN, который является модификацией алгоритма SMOTE. Авторы приводят примеры, которые показывают, что улучшенный алгоритм дает более качественные результаты, чем алгоритм SMOTE на пяти множествах.

В публикации [24] рассматривается задача классификации твитов (записей в микроблоге Twitter) по социальному окрасу. Рассматривается качество работы классификаторов при разной степени дисбаланса классов, а также при использовании алгоритма Random Undersampling, который уменьшает количество элементов в доминирующем классе до числа, равному количеству элементов в минорном классе. Было выявлено, что на данных, к которым был применен алгоритм, производительность алгоритма существенно выше, чем при работе на необработанных данных.

Сайт [5] является документацией к реляционной системе управления базами данных (РСУБД) MySQL.

Также, в списке литературы присутствуют документы, которые являются документацией к программным пакетам на языке Python, которые использовались в данной работе:

- [9] MySQLdb – библиотека, предоставляющий доступ к РСУБД MySQL;
- [10] Matplotlib – библиотека для визуализации данных, построения графиков, диаграмм и т. д.;
- [19] Scikit-learn – библиотека для работы с методами машинного обучения
- [25] Imbalanced-learn – библиотека для работы с несбалансированными выборками.

Глава 1. Обзор МИС «Виста-Мед»

1.1. Обзор графического интерфейса приложения

МИС «Виста-Мед» представляет из себя веб-приложение с графическим интерфейсом. Она позволяет вести всю документацию лечебного учреждения в электронном виде. Каждый врач имеет собственный аккаунт с определенными правами доступа. Ему предоставляются медицинские карты пациентов, направленных к нему. В электронную медицинскую карту заносятся сведения об анализах, назначенных процедурах, дневники осмотров, температурные листы, рецепты, диагнозы и эпикризы. Имеется список пациентов по палатам, также в каждой медицинской карте записаны личные (ФИО) и паспортные данные, полис ОМС. Кроме того, система позволяет создавать необходимые отчеты и счета для страховых компаний.

На Рисунке 1 представлен графический интерфейс дневника осмотров. В него заносятся жалобы и симптомы. Слева отображаются общие сведения о пациенте (личные данные, аллергии, диагнозы, тип финансирования и хронология движения пациента по отделениям). Справа отображаются список проведенных осмотров и процедур с указанием даты и времени.

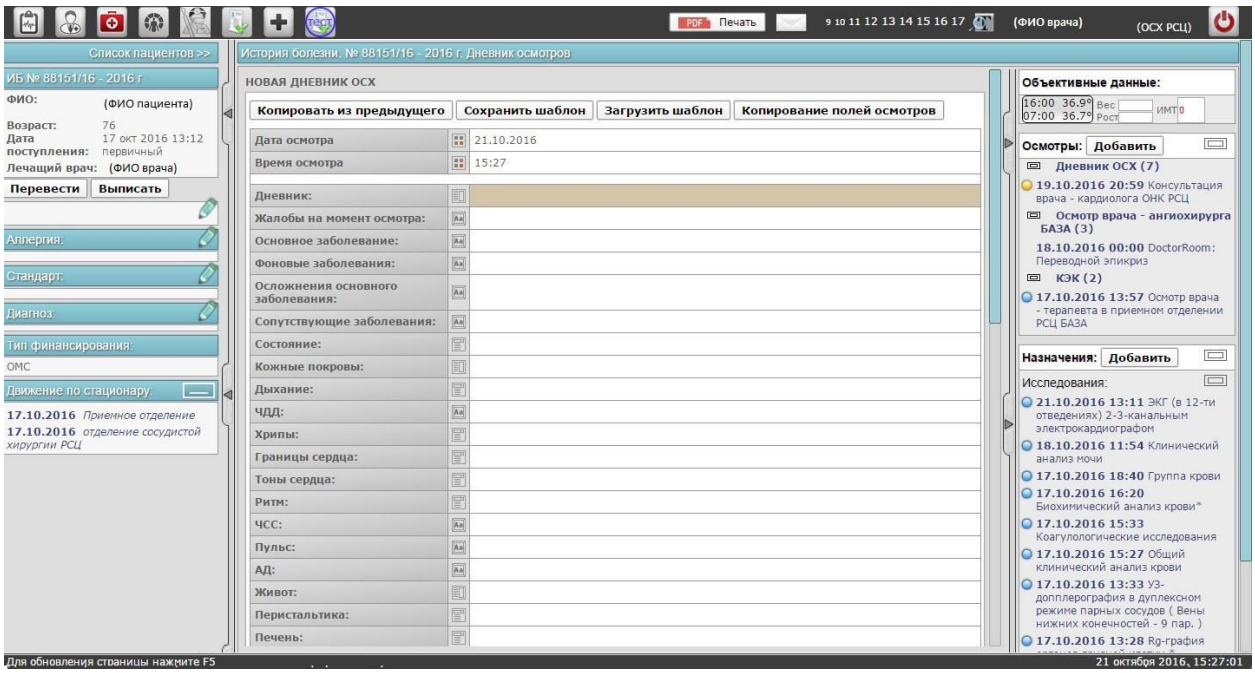


Рисунок 1. Дневник осмотров и медицинская карта пациента.

Щелкнув на проведенный анализ или осмотр, врач может узнать его результаты. Показатели, выходящие за пределы нормы выделяются красным цветом. На Рисунке 2, например, можно наблюдать результаты биохимического анализа крови.

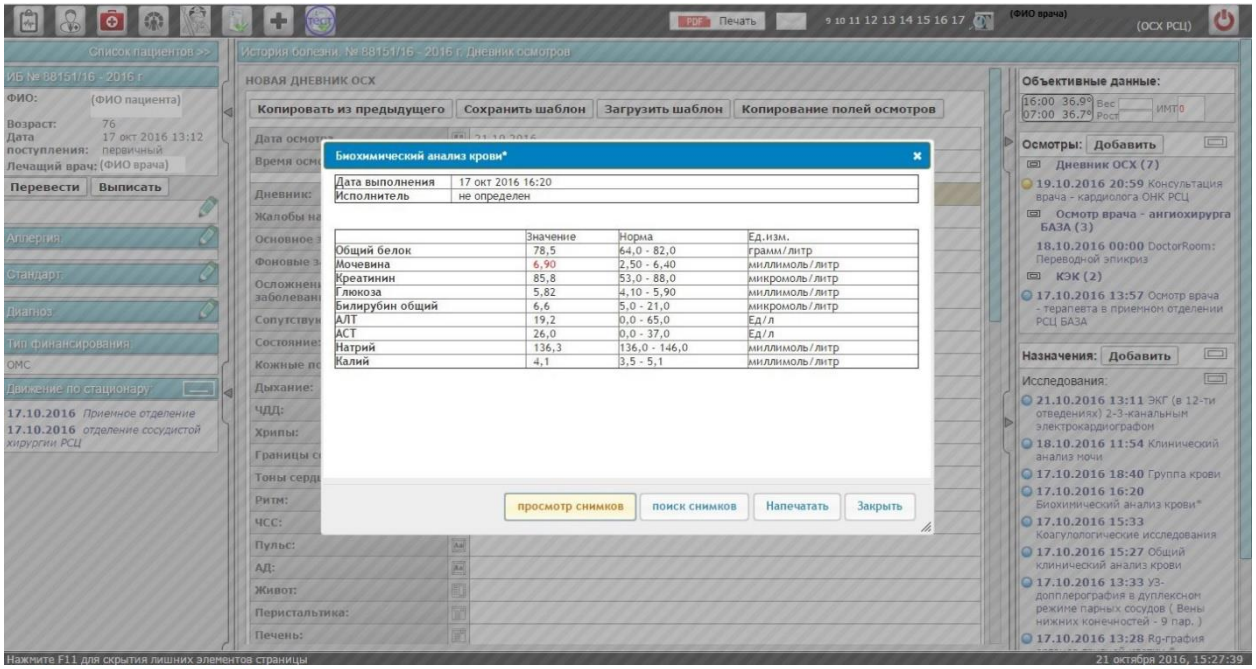


Рисунок 2. Данные о проведенном анализе.

Больше скриншотов можно найти в приложении 1.

Также имеется аналогичное десктопное приложение.

1.2. Обзор архитектуры баз данных приложения

Управление данными в МИС «Виста-мед» осуществляется с помощью СУБД (система управления базами данных) «Microsoft SQL Server». В базе находится 501 таблица, включая справочные таблицы и таблицы, оставшиеся от предыдущих версий программы. Из них таблиц и полей в них, которые содержат действительно необходимую для нас информацию не так уж и много. Рассмотрим их.

Итак, все клиенты записываются в таблицу Client. В ней хранятся личные данные пациента (см. Таблицу 1).

Таблица 1. Структура таблицы Client.

Название поля	Тип данных	Примечание
id	int	Первичный ключ
lastName	varchar	Фамилия
firstName	varchar	Имя
patrName	varchar	Отчество
birthdate	date	Дата рождения
sex	tinyint	Пол

Каждое посещение клиентом учреждения фиксируется в таблице Event (см. Таблицу 2).

Таблица 2. Структура таблицы Event.

Название поля	Тип данных	Примечание
id	int	Первичный ключ
client_id	int	Внешний ключ на таблицу Client
setDate		Дата посещения пациентом

Все диагнозы, которые ставятся пациентам заносятся в таблицу Diagnosis (см. Таблицу 3).

Таблица 3. Структура таблицы Diagnosis.

Название поля	Тип данных	Примечание
id	int	Первичный ключ
client_id	int	Внешний ключ на таблицу Client

MKB	varchar	Код заболевания по МКБ

Расшифровка всех кодов МКБ находится в справочной таблице MKB (см Таблицу 4). Подробнее об этом рассказано в Главе 2.

Таблица 4. Структура таблицы MKB.

Название поля	Тип данных	Примечание
id	int	Первичный ключ
ClassName	varchar	Класс заболеваний
DiagID	varchar	Индекс по МКБ
DiagName	varchar	Наименование болезни

Привязка диагноза к посещению происходит через таблицу Diagnostic (см. Таблицу 5).

Таблица 5. Структура таблицы Diagnostic.

Название поля	Тип данных	Примечание
id	int	Первичный ключ
event_id	int	Фамилия
diagnosis_id	int	Внешний ключ на таблицу Diagnosis
diagnosisType_id	int	Тип диагноза (внешний ключ на таблицу rbDiagnosisType)

Типы диагнозов описываются в справочной таблице rbDiagnosisType (см. Таблицу 6).

Таблица 6. Структура таблицы rbDiagnosisType.

Название поля	Тип данных	Примечание
id	int	Первичный ключ
name	varchar	Наименование типа диагноза

Все возможные типы действий, осуществляемых с пациентом, фиксируются в таблице ActionType (см. Таблицу 7).

Таблица 7. Структура таблицы ActionType.

Название поля	Тип данных	Примечание
id	int	Первичный ключ
name	varchar	Наименование действия
serviceType	tinyint	Вид услуги: 0 - Прочие, 1 - первичный осмотр, 2 - повторный осмотр, 3 - процедура/манипуляция, 4 -

		операция, 5 - исследование, 6 - лечение

В свою очередь, действия, осуществляемые с конкретным пациентом записываются в таблицу Action (см. Таблицу 8).

Таблица 8. Структура таблицы Action.

Название поля	Тип данных	Примечание
id	int	Первичный ключ
actionType_id	int	Внешний ключ на таблицу ActionType
event_id	int	Внешний ключ на таблицу Event

Некоторые типы действий имеют свойства. Если они имеются, то они указываются в таблице ActionPropertyType (см. Таблицу 9).

Таблица 9. Структура таблицы ActionPropertyType.

Название поля	Тип данных	Примечание
id	int	Первичный ключ
actionType_id	int	Внешний ключ на таблицу ActionType
name	varchar	Наименование свойства действия

Для конкретных действий, сведения о наличии того или иного свойства, записаны в таблицу ActionProperty (см. Таблицу 10).

Таблица 10. Структура таблицы ActionProperty.

Название поля	Тип данных	Примечание
id	int	Первичный ключ
action_id	int	Внешний ключ на таблицу Action
type_id	int	Внешний ключ на таблицу ActionPropertyType

Числовые и символьные значения свойств конкретных действий, записываются в таблицу ActionProperty_String (см. Таблицу 11).

Таблица 11. Структура таблицы ActionProperty_String.

Название поля	Тип данных	Примечание
id	int	Внешний ключ на таблицу ActionProperty
value	varchar	Значение свойства действия

Также, каждая таблица содержит поле deleted. Оно используется для удаления данных в графическом интерфейсе МИС. Значение этого поля равно 1, если запись была удалена, и 0 иначе.

Для наглядности была построена диаграмма базы данных с

использованием соответствующих инструментов в MySQL Workbench [5] (см. Рисунок 3).

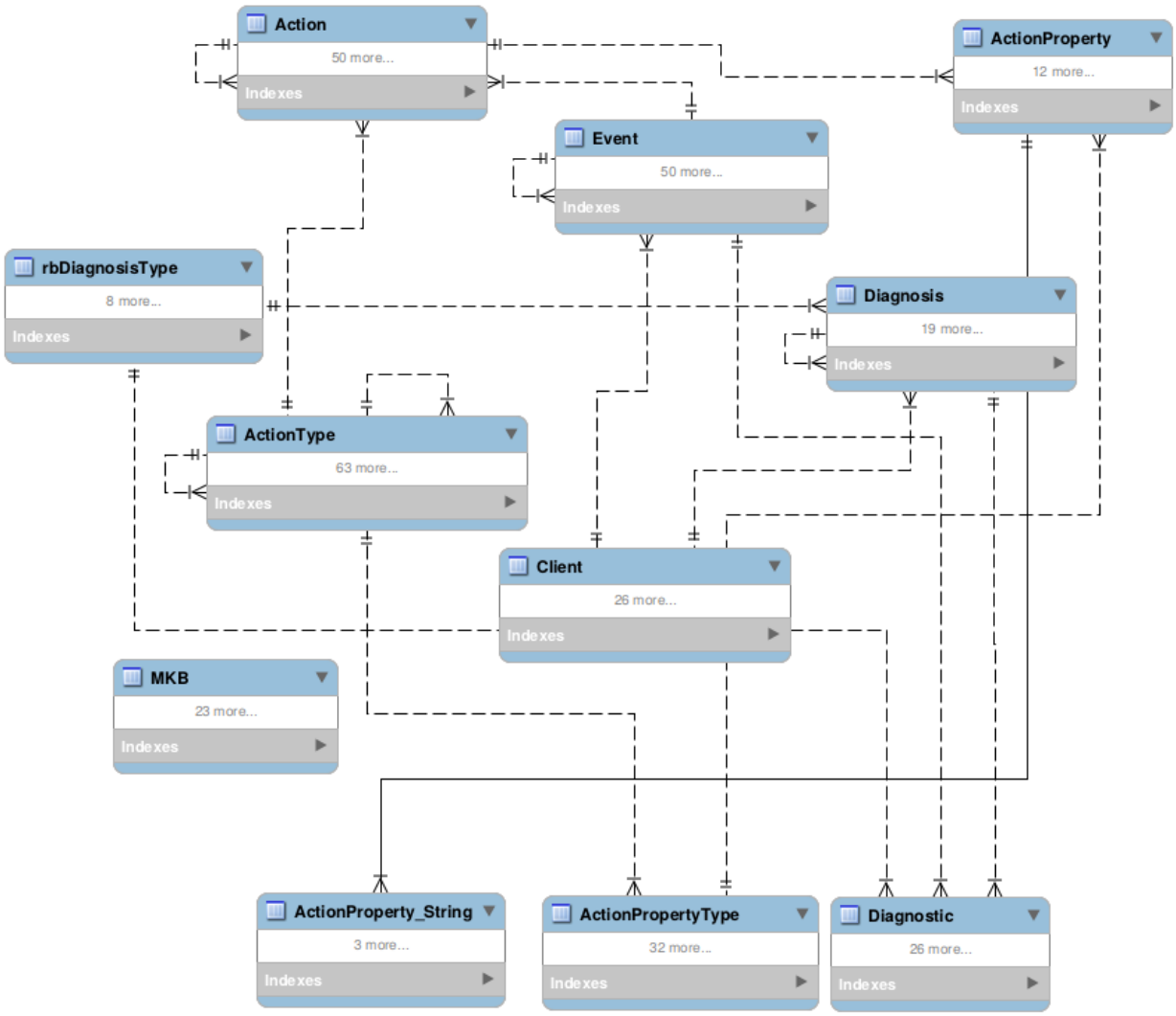


Рисунок 3. Диаграмма баз данных МИС «Виста-Мед».

Таким образом, механизм записи данных при каждом посещении пациента работает следующим способом:

1. Заносятся сведения о пациенте в таблицу Client (если он посещает учреждение первый раз);

2. Создается запись в таблице Event о посещении пациентом заведения с привязкой к конкретному клиенту;

3. Все действия, которые происходят с пациентом записываются в таблицу Action с привязкой к конкретному посещению;

4. Сведения о наличии тех или иных свойств заносятся в таблицу ActionProperty с привязкой к конкретному действию. Значения числовых и

символьных свойства записываются в таблицу ActionProperty_String с привязкой к конкретному свойству конкретного действия. Если пациенту во время посещения заведения ставится какой-либо диагноз, то соответствующие сведения об этом заносятся в таблицы Diagnosis и Diagnostic с привязкой к конкретному посещению пациента.

Глава 2. Создание классификатора, принимающего решение о необходимости операции пациенту

2.1. Подготовка данных, создание выборки, выбор признаков

Итак, нам был дан дамп базы данных SQL в формате .sql. Для его развертывания использовалась РСУБД (реляционная система управления базами данных) MySQL версии 5.7 и инструмент для визуализации MySQL Workbench [5]. После развертывания мы смогли работать непосредственно с базой и делать к ней запросы.

Перед нами стоит задача классификации, которая относится к категории задач обучения с учителем. Нам дано множество всех возможных объектов X, и множество допустимых классов Y. Элемент множества X представляют из себя вектор размерности n из признаков, которые представляют из себя характеристику некоторого свойства объекта. Признак может быть бинарным, порядковым (категориальным), количественным и т. д.

В свою очередь, известна некоторая целевая функция y^*, которая является отображением вида $X \to Y$ и известны ее значения на ограниченном подмножестве $(x_1, \dots, x_l) \subset X$ множества X. Совокупность пар объект – значение $X^l = (x_i, y_i)_{i=1}^l$ называется тренировочной выборкой.

Задача классификации заключается в том, чтобы построить по тренировочной выборке X^l решающую функцию a, которая будет являться приближением целевой функции y^* на всем множестве X [6].

В нашей конкретной задаче объектами являлись пациенты, которые делятся на 2 класса: пациенты, которым назначена операция и пациенты, которым операцию не назначили. Задача классификации, в которой количество допустимых классов равно двум называется задачей бинарной классификации [6].

Для начала необходимо было выбрать наиболее распространённое

заболевание среди пациентов. В будущем, больший размер обучающей выборки позволит повысить качество обучения классификатора.

Заболевания в МИС «Виста-Мед» характеризуются с помощью МКБ – международной классификации болезней. Каждая болезнь унифицируется уникальным трёхзначным кодом. Периодически МКБ пересматривается, вносятся изменения, исправляются ошибки. Сейчас действует МКБ-10 (международная классификация болезней десятого пересмотра), принятая Всемирной организацией здравоохранения [7].

Все заболевания в МКБ-10 разделены на классы, всего их 22. Индексы заболеваний в каждом классе начинаются с определенной латинской буквы. Так как нам даны данные с сердечно-сосудистого отделения больницы, то заболевания пациентов относятся к классу IX «Заболевания системы кровообращения». С полным списком классов заболеваний можно ознакомиться в книге [8].

Индексы заболеваний класса IX начинаются с буквы «I». Таким образом, чтобы вывести все случаи, в которых болезни с сердечно-сосудистыми заболеваниями фигурировали как основные, необходимо будет выполнить следующий SQL-запрос:

```
«SELECT b.id, b.event_id, b.diagnosisType_id FROM
                    Diagnosis AS a
  JOIN Diagnostic AS b ON a.id = b.diagnosis_id
WHERE a.MKB LIKE 'I%' AND b.diagnosisType_id IN (1,
    2, 12) AND a.deleted = 0 AND b.deleted = 0»
```

Поле diagnosisType_id должно принимать значение 1, 2 или 12, так как эти значения соответствуют определенным типам диагнозов (1 - заключительный диагноз, 2 - основной, 12 - основной предварительный).

Для сбора данных и статистики использовался язык программирования Python версии 3.5 и библиотеки MySQLdb [9] для работы с СУБД и Matplotlib [10] для построения диаграмм. Получаем следующее распределение пациентов по болезням (см. Таблицу 12, Рисунок 4).

Таблица 12. Распределение пациентов по болезням.

Код заболевания (МКБ)	**Расшифровка**	**Количество пациентов**
I63	Инфаркт мозга	79
I20	Стенокардия [грудная жаба]	40
I82	Эмболия и тромбоз других вен	31
I21	Острый инфаркт миокарда	27
I80	Флебит и тромбофлебит	25
I61	Внутримозговое кровоизлияние	23
I11	Гипертензивная болезнь сердца [гипертоническая болезнь с преимущественным поражением сердца]	19
I70	Атеросклероз	18
I64	Инсульт, не уточненный как кровоизлияние или инфаркт	17
I25	Хроническая ишемическая болезнь сердца	16
I67	Другие цереброваскулярные болезни	15
I83	Варикозное расширение вен нижних конечностей	9
I23	Некоторые текущие осложнения острого инфаркта миокарда	6
I74	Эмболия и тромбоз артерий	4
I69	Последствия цереброваскулярных болезней	4
I47	Пароксизмальная тахикардия	4
I60	Субарахноидальное кровоизлияние из каротидного синуса и бифуркация	3
I50	Сердечная недостаточность	2
I89	Другие неинфекционные болезни лимфатических сосудов и лимфатических узлов	2
I24	Другие формы острой ишемической болезни сердца	2
I12	Гипертензивная [гипертоническая] болезнь с преимущественным поражением почек	2
I34	Неревматические поражения митрального клапана	1
I10	Эссенциальная [первичная] гипертензия	1
I01	Ревматическая лихорадка с вовлечением сердца	1
I65	Закупорка и стеноз прецеребральных артерий, не приводящие к инфаркту мозга	1

I49	Другие нарушения сердечного ритма	1
I71	Аневризма и расслоение аорты	1
I30	Острый перикардит	1
I45	Другие нарушения проводимости	1
I62	Другое нетравматическое внутричерепное кровоизлияние	1
I79	Поражения артерий, артериол и капилляров при болезнях, классифицированных в других рубриках	1

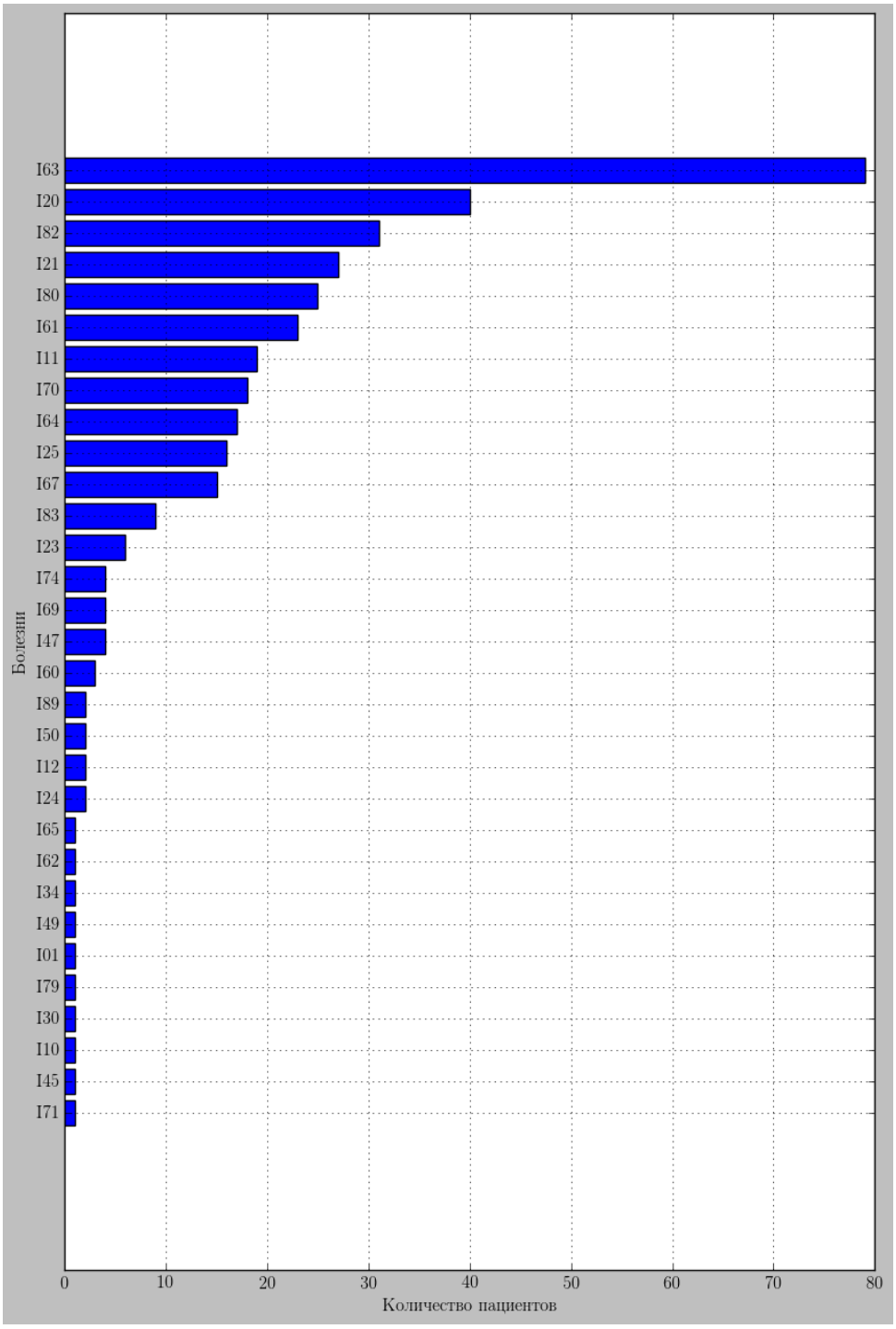


Рисунок 4. Распределение пациентов по болезням.

Так как мы заинтересованы в том, чтобы размер выборки был как можно больше, создадим выборку из пациентов, имеющих заболевание I63 – инфаркт мозга.

Инфаркт мозга возникает по причине недостаточного кровоснабжения определенной части мозга, тромбоза или эмболии. Поэтому, в качестве признаков будем использовать данные общего клинического анализа крови. Он включает в себя 13 числовых характеристик:

1. Средняя концентрация гемоглобина в эритроцитах.
2. Гемоглобин.
3. Коэффициент анизотропии эритроцитов.
4. Эритроциты.
5. Лимфоциты.
6. Среднее содержание гемоглобина.
7. Процент лимфоцитов.
8. Гематокрит.
9. Лейкоциты.
10. Тромбоциты.
11. Средний объем тромбоцита.
12. Средний объем эритроцита.
13. Цветовой показатель.

Для того чтобы узнать для каждого пациента этот набор признаков и была ли сделана тому или иному пациенту операция, необходимо будет найти все действия, выполненные с ним. Для этого нужно выполнить следующий запрос:

```
"SELECT a.id, a.actionType_id, b.serviceType FROM
Action AS a JOIN ActionType AS b ON a.actionType_id =
b.id WHERE event_id = %s"
```

Если находим хотя бы одно действие, у которого serviceType = 4, значит данному пациенту совершалась операция. Иначе, пациенту не была произведена операция.

Далее, среди всех действий находим общий клинический анализ крови (actionType_id = 26571) для каждого пациента. Для этого составляем запрос:

```
"SELECT c.id, b.name, c.value FROM ActionProperty
as a JOIN ActionPropertyType AS b ON a.type_id = b.id
JOIN ActionProperty_String AS c ON a.id = c.id
WHERE action_id = %s"
```

Берутся данные самого первого анализа, до начала лечения, чтобы учитывались признаки еще не вылеченного пациента.

2.2. Алгоритмы классификации

Для создания классификаторов были выбраны следующие методы: SVM (support vector machine - метод опорных векторов), дерево принятия решений и random forest (случайный лес). Рассмотрим их подробнее.

2.2.1. Метод опорных векторов

Метод был предложен в 60-70-е годы советским математиком В. Н. Вапником. SVM считается одним из лучших методов классификации на данный момент.

Представим элементы выборки как точки на плоскости размерности n, где n – количество признаков. Часть из них уже размечена (тренировочное множество) и принадлежит либо классу 0, либо классу 1, а часть еще предстоит разметить (тестовое множество). Метод опорных векторов заключается в построении оптимальной гиперплоскости, разделяющей объекты 0 и 1 классов, при условии, что ближайшие размеченные объекты должны быть максимально удалены от нее (линейно разделимая выборка). Именно эти объекты и называются опорными векторами. Чем больше будет зазор между двумя классами, тем надежнее должен быть полученный классификатор. Если уравнение гиперплоскости будет иметь вид $wx - b = 0$, где w – вектор нормали. Расстояние от оптимальной гиперплоскости до ближайшего объекта любого из классов равно $\frac{1}{w}$, следовательно ширина разделяющей полосы будет

равна $\frac{2}{w}$. Следовательно, наша задача заключается в максимизации ширины и полосы, т. е. минимизации нормы вектора w (см. Рисунок 5).

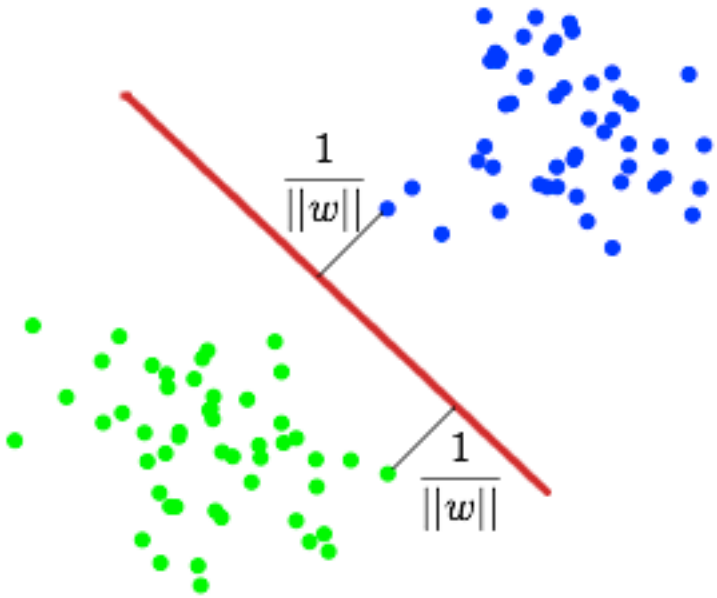


Рисунок 5. Построение оптимальной разделяющей гиперплоскости.

Если классы линейно неразделимы, то вводятся переменные ошибки ξ_i, $i = \overline{1,\ l}$ (l – количество объектов в обучающей выборке) и в условие минимизации w добавляется штрафная функция.

Помимо линейного разделения, может осуществляться разделение может осуществляться нелинейно. Для этого вводится функция ядра $K(x, x')$. В этой работе будут использоваться следующие ядра:

- Линейное: $K(x, x') = \langle x, x' \rangle$ (скалярное произведение).
- Полиномиальное:$K(x, x') = \langle x, x' \rangle^d$.
- Сеть радиально-базисных функций (RBF): $K(x, x') = \exp(-\gamma \|x - x'\|^2)$.

Подробнее об методе опорных векторов можно прочитать в материалах [6, 11].

2.2.2. Дерево принятия решений

Дерево решений относится к логическим методам классификации (является композицией интерпретируемых закономерностей). Деревом в теории графов называется связный неориентированный граф, при условии отсутствия в нем циклов [12]. Вершины, из которых не выходят ребра называются листьями, все остальные вершины называются внутренними.

В данном случае будут использоваться бинарные деревья решений. В их основе стоит бинарные деревья, внутренние ребра которых имеют левое дочернее и правое дочернее ребра. При этом каждая внутренняя вершина v, принадлежащая дереву V является предикатом $\beta_v: X \to \{0, 1\}$, а каждому листу v, принадлежащему дереву V приписан определенный класс [13]. Пример дерева решений можно наблюдать на рисунке 6.

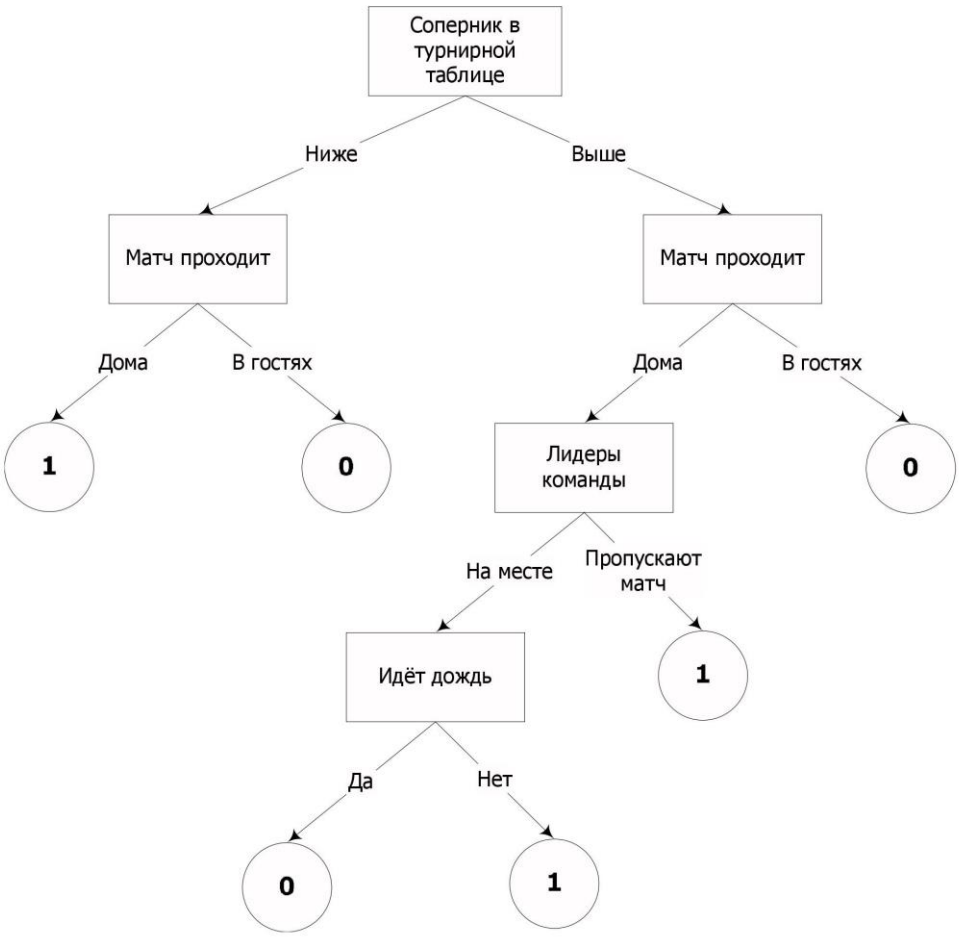


Рисунок 6. Пример бинарного дерева решений.

Таким образом, каждый объект из тестового множества проходит дерево от корня до какого-либо из листьев, где ему и присваивается класс.

Существует несколько правил построения таких деревьев: ID3, C4.5, CART и другие. В программном пакете, используемом мной реализован метод CART (Classification and Regression Tree), который был применен для решения задачи. Он предназначен для построения бинарных деревьев решений. Его основной смысл заключается в том, что в процессе создания дерева выполняется рекурсивное разбиение примеров обучающего множества на подмножества, объекты в которых принадлежат одному и тому же классу.

Пусть множество A содержит в себе n элементов, из них m элементов обладают некоторым свойством S, которое принимает значение s для m_i элементов. Для оценки качества полученного дерева могут использоваться два критерия:

- Индекс Джини $Gini(A,S) = 1 - \sum_{i=1}^{s} \frac{mi}{n}$;
- Энтропийный (прироста информации) $H(A,S) = -\sum_{i=1}^{s} \frac{m_i}{n} \log \frac{m_i}{n}$.

В процессе создания дерева алгоритм CART проводит для каждого узла полный перебор всех свойств (признаков), на основе которых может быть выполнено разбиение, и выбирает тот, который максимизирует значение выбранного критерия.

Часто деревья получаются слишком подробными. Для того, чтобы избежать явления переобучения прибегают к отсечению лишних ветвей (pruning) [13,14,15].

2.2.3. Случайный лес

Алгоритм Random forest (случайный лес) был предложен в 1995 году Тин Кам Хо и усовершенствован Лео Брейманом в 2001. Он представляет собой ансамбль из n решающих деревьев, для каждого из которых случайно выбираются θ случайных объектов из обучающего множества. Далее деревья обучаются на этих объектах, и методом голосования тому или иному элементу тестового множества присваивается определенный класс. Эксперименты Бреймана показали, что этот алгоритм работает лучше, чем одно решающее дерево [16].

2.3. Меры качества классификации и результаты при кросс-валидации

Для оценивания качества полученного бинарного классификатора существуют меры, позволяющие оценить его работу. В данной работе использовались основные качественные меры классификации: точность (precision), полнота (recall), f-мера, правильность (accuracy). Рассмотрим их подробнее.

Итак, имеем обученный бинарный классификатор. Разделим принятые им решения на 4 категории:

- Истинно положительные решения – случаи, при которых классификатор распознал правильно положительный класс;
- Ложно положительные решения – случаи, при которых классификатор неправильно распознал положительный класс;
- Истинно отрицательные решения – случаи, при которых классификатор правильно распознал отрицательный класс;
- Истинно положительные решения – случаи, при которых классификатор правильно распознал отрицательный класс (см. Таблицу 13).

Таблица 13. Таблица сопряженных признаков.

		Истинная оценка	
		Положительная	Отрицательная
Оценка системы	Положительная	Истинно положительные решения (true positive – tp)	Ложно положительные решения (false positive - fp)
	Отрицательная	Ложно отрицательные решения (false negative – fn)	Истинно отрицательные решения (true negative – tn)

Теперь введем определения метрик качества.

Точность (precision) – это доля правильно предсказанных объектов положительного класса среди всех объектов, предсказанных как положительные, т.е.

$$precision = \frac{tp}{tp + fp}$$

Полнота (recall) – это доля правильно предсказанных объектов, среди всех объектов положительного класса, т. е.

$$recall = \frac{tp}{tp + fn}$$

F–мера – это показатель, который позволяет найти баланс между точностью и полнотой и объединяющий информацию о них. Она находится как среднее гармоническое между точностью и полнотой, т. е.

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

Правильность (accuracy) – это доля правильно предсказанных объектов среди всей выборки, т.е.

$$Accuracy = \frac{tp + tn}{tp + fp + fn + tn}$$

Чаще всего, при оценке работы классификатора используется правильность. Но она не всегда подходит для оценки, так как очень часто данные в реальных ситуациях несбалансированные, и одной правильности может быть недостаточно для объективной оценки [17].

Так как размер нашей выборки очень мал (всего лишь 79 элементов), для объективной оценки классификатора придется использовать кросс-валидацию. Она заключается в следующем:

- выборка делится на k частей;
- $k - 1$ часть используется для обучения модели;
- последняя часть используется для проверки качества модели.

Данный алгоритм повторяется k раз и точность модели находится как

среднее арифметическое всех полученных значений на k шагах. Кросс-валидация используется, чтобы избежать переобучения классификатора и правильно оценить полученную модель машинного обучения [18].

Для создания моделей были использованы язык программирования Python версии 3.5 и библиотека для использования методов машинного обучения Scikit-learn [19]. Библиотека содержит в себе большое количество методов и удобные инструменты для применения кросс-валидации и оценки моделей машинного обучения. Были получены следующие результаты при использовании кросс-валидации с коэффициентом $k = 3$ ($\frac{2}{3}$ выборки – тренировочное множество и $\frac{1}{3}$ – тестовое) и использовании метода опорных векторов, деревьев решений и случайного леса (см. Таблицу 14):

Таблица 14. Оценки классификаторов, принимающих решение о необходимости операции.

Метод классификации	**precison**	**recall**	**f-мера**	**accuracy**
SVM (с линейным ядром)	0	0	0	0,9391
SVM (с полиномиальным ядром)	0.3333	0.3333	0.3333	0,8897
SVM (с RBF ядром)	0	0	0	0,9634
Дерево решений (по критерию Джини с максимальной глубиной 3 и более)	0	0	0	0,9025
Дерево решений (по энтропийному критерию с максимальной глубиной 3 и более)	0	0	0	0,8902
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 3 и более)	0	0	0	0,9634
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 3 и более)	0	0	0	0,9634
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 3 и более)	0	0	0	0,9634
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 3 и более)	0	0	0	0,9634

По высокой правильности можно подумать, что полученный

классификатор является качественным. Но как можно наблюдать, значения точности и полноты очень малы, а значит, результаты оставляют желать лучшего. Эмпирическим путем было выяснено, полученная выборка несимметрична, и в ней содержится только 3 объекта, которым была сделана операция (положительный класс), а всем остальным 76 пациентам не была сделана операция (отрицательный класс). Вследствие этого, классификатор очень хорошо предугадывает объекты отрицательного класса, но не справляется с определением объектов, принадлежащих положительному классу. Если пациент, которому требуется операция, не будет прооперирован это может негативно сказаться на его здоровье и возможно, привести к летальному исходу.

Решение данной проблемы будет предложено в Главе 4.

Глава 3. Создание классификатора, определяющего болезнь пациента

3.1. Создание выборки, выбор признаков

Безусловно, постановка операции тому или иному пациенту играет важную роль в его лечении. Но еще более важно, поставить ему правильный диагноз, чтобы назначить для пациента необходимое лечение. Ведь неправильная диагностика болезни может только усугубить пагубное влияние на организм, и как следствие, привести к летальному исходу. Поэтому, было принято решение обобщить нашу задачу до создания классификатора, определяющего болезнь пациента.

Так как размер классов по многим болезням очень мал (см. Таблицу 12), было принято решение включить в выборку, только те болезни, по которым есть 20 и более пациентов (см. Таблицу 15, Рисунок 7).

Таблица 15. Распределение болезней с количеством пациентов 20 и более.

Код заболевания (МКБ)	**Расшифровка**	**Количество пациентов**
I63	Инфаркт мозга	79
I20	Стенокардия [грудная жаба]	40
I82	Эмболия и тромбоз других вен	31
I21	Острый инфаркт миокарда	27
I80	Флебит и тромбофлебит	25
I61	Внутримозговое кровоизлияние	23

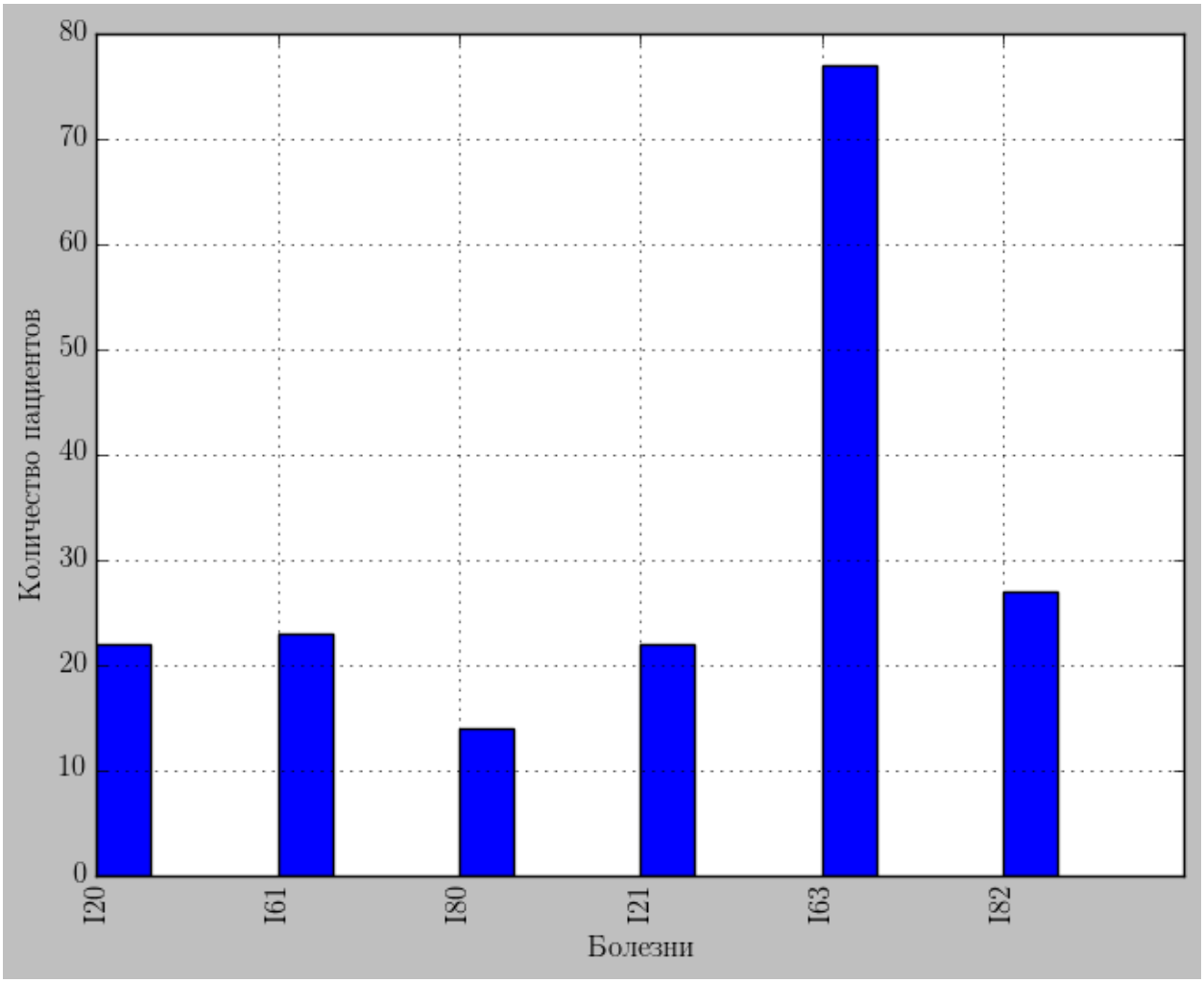


Рисунок 7. Распределение болезней с количеством пациентов 20 и более.

Для начала, попробуем взять в качестве признаков классификации все тот же общий клинический анализ крови. Для этого были осуществлены аналогичные запросы, которые были предназначены для создания выборки в главе 2, но не находилось значение поля serviceType для всех действий, выполненных с пациентом, и область значений целевой функции представляет из себя список заболеваний пациентов (см. выше).

3.2. Применение алгоритмов классификации и результаты при кросс-валидации

Размер классов и выборки в целом невелики (см. Таблицу 12), поэтому в данном случае тоже будет использовать кросс-валидацию. В качестве показателя качества полученной модели будет использоваться только правильность (accuracy). Так как классы не являются такими

несбалансированными, как в задаче бинарной классификации в главе 2, ее будет достаточно. Для многоклассовой классификации она рассчитывается следующим образом:

$$accuracy = \frac{P}{N},$$

где P – количество правильно угаданных объектов из тестовой выборки, а N – размер тестовой выборки.

При использовании метода опорных векторов, дерева решений и случайного леса были получены следующие результаты (см. Таблицу 16):

Таблица 16. Оценки классификаторов, определяющих болезнь пациента по основному клиническому анализу крови.

Метод классификации	**Правильность**
SVM (с линейным ядром)	0,2476
SVM (с полиномиальным ядром)	0,2547
SVM (с RBF ядром)	0,4164
Дерево решений (по критерию Джини с максимальной глубиной 3)	0,2373
Дерево решений (по критерию Джини с максимальной глубиной 4)	0,2368
Дерево решений (по критерию Джини с максимальной глубиной 5)	0,2193
Дерево решений (по критерию Джини с максимальной глубиной 6)	0,2640
Дерево решений (по критерию Джини с максимальной глубиной 7)	0,3012
Дерево решений (по критерию Джини с максимальной глубиной 8)	0,3276
Дерево решений (по критерию Джини с максимальной глубиной 9)	0,2840
Дерево решений (по критерию Джини с максимальной глубиной 10)	0,2953
Дерево решений (по критерию Джини с максимальной глубиной 11)	0,3163
Дерево решений (по критерию Джини с максимальной глубиной 12 и более)	0,2903
Дерево решений (по энтропийному критерию с максимальной глубиной 3)	0,1877
Дерево решений (по энтропийному критерию с максимальной глубиной 4)	0,1891
Дерево решений (по энтропийному критерию с максимальной глубиной 5)	0,2534
Дерево решений (по энтропийному критерию с максимальной глубиной	0,2861

6)	
Дерево решений (по энтропийному критерию с максимальной глубиной 7)	0,2854
Дерево решений (по энтропийному критерию с максимальной глубиной 8)	0,2803
Дерево решений (по энтропийному критерию с максимальной глубиной 9)	0,2756
Дерево решений (по энтропийному критерию с максимальной глубиной 10 и более)	0,2595
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 3)	0,3026
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 4)	0,3241
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 5)	0,3612
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 6)	0,3942
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 7)	0,4044
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 8)	0,3945
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 9)	0,3993
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 10)	0,3777
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 11)	0,3886
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 12)	0,3830
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 13 и более)	0,3988
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 3)	0,2584
Случайный лес (с 20 деревьями по энтропийному критерию с	0,3079

максимальной глубиной 4)	
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 5)	0,3714
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 6)	0,3675
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 7)	0,3772
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 8)	0,4165
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 9)	0,4003
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 10)	0,3889
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 11 и более)	0,3942
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 3)	0,3399
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 4)	0,3354
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 5)	0,3614
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 6)	0,3734
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 7)	0,3900
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 8)	0,3724
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 9)	0,4109
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 10)	0,3568
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 11)	0,3896
Случайный лес (с 10 деревьями по критерию Джини с максимальной	0,3843

глубиной 12)	
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 13 и более)	0,3948
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 3)	0,2277
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 4)	0,3181
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 5)	0,3660
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 6)	0,3244
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 7)	0,3938
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 8)	0,4161
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 9)	0,4331
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 10)	0,4052
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 11 и более)	0,4108

Наилучший результат для данной выборки дал алгоритм случайного леса при использовании 10 деревьев по энтропийному критерию с максимальной глубиной дерева, равной 9 – 0,4331. Как можно заметить, в некоторых случаях ограничение глубины деревьев способствует увеличению правильности при применении дерева решений и случайного леса. Также неплохо показал себя метод опорных векторов с RBF ядром – 0,4164. Это уже лучше, чем константный классификатор, однако, мы заинтересованы в улучшении работы модели, поэтому попробуем добавить или использовать другие признаки объектов, которые, возможно положительно повлияют на качество модели машинного обучения.

3.3. Результаты при учете других признаков пациента

3.3.1. Добавление возраста к признакам

Часто, те или иные болезни характерны для определенных возрастных категорий. Попробуем к основному клиническому анализу крови добавить возраст пациента.

Чтобы узнать его, необходимо найти разницу между полем setDate (дата посещения пациента) в таблице Event и полем birthDate в таблице Client (дата рождения пациента). Для нахождения разницы между датами в днях в MySQL существует функция DATEDIFF() [9]. Переведем дни в формат date и найдем количество лет. Таким образом, получаем следующий SQL-запрос:

```
"SELECT
YEAR(FROM_DAYS(DATEDIFF(DATE(e.setDate),c.birthDate)))
FROM Client AS c JOIN Event AS e ON e.client_id = c.id
WHERE e.id = %s"
```

Добавим возраст пациента к характеристикам основного анализа крови. Теперь, каждый объект выборки представляет из себя 14-ти компонентный вектор. Посмотрим, как поведут себя ранее использованные методы классификации на данной выборке в сравнении с выборкой пациентов, в которой их возраст не учитывается (см. Таблицу 17).

Таблица 17. Оценки классификаторов, определяющих болезнь пациента по основному клиническому анализу крови и возрасту.

Метод классификации	**Правильность**
SVM (с линейным ядром)	0,2476
SVM (с полиномиальным ядром)	0,2649
SVM (с RBF ядром)	0,4164
Дерево решений (по критерию Джини с максимальной глубиной 3)	0,2525
Дерево решений (по критерию Джини с максимальной глубиной 4)	0,2203
Дерево решений (по критерию Джини с максимальной глубиной 5)	0,2468
Дерево решений (по критерию Джини с максимальной глубиной 6)	0,2843
Дерево решений (по критерию Джини с максимальной глубиной 7)	0,3046

Дерево решений (по критерию Джини с максимальной глубиной 8)	0,3380
Дерево решений (по критерию Джини с максимальной глубиной 9)	0,3264
Дерево решений (по критерию Джини с максимальной глубиной 10)	0,3166
Дерево решений (по критерию Джини с максимальной глубиной 11)	0,3219
Дерево решений (по критерию Джини с максимальной глубиной 12 и более)	0,3272
Дерево решений (по энтропийному критерию с максимальной глубиной 3)	0,1947
Дерево решений (по энтропийному критерию с максимальной глубиной 4)	0,1838
Дерево решений (по энтропийному критерию с максимальной глубиной 5)	0,2536
Дерево решений (по энтропийному критерию с максимальной глубиной 6)	0,2861
Дерево решений (по энтропийному критерию с максимальной глубиной 7)	0,2967
Дерево решений (по энтропийному критерию с максимальной глубиной 8)	0,3177
Дерево решений (по энтропийному критерию с максимальной глубиной 9 и более)	0,3073
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 3)	0,2814
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 4)	0,3619
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 5)	0,3828
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 6)	0,3556
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 7)	0,3607
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 8)	0,4259
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 9)	0,4422

Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 10)	0,4151
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 11)	0,4162
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 12)	0,4108
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 13)	0,4214
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 14 и более)	0,4162
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 3)	0,3135
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 4)	0,3771
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 5)	0,3781
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 6)	0,3885
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 7)	0,4321
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 8)	0,4050
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 9)	0,4038
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 10)	0,4104
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 11 и более)	0,4047
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 3)	0,2156
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 4)	0,3180
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 5)	0,3773

Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 6)	0,3183
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 7)	0,3611
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 8)	0,4057
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 9)	0,4271
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 10)	0,4104
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 11)	0,4324
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 12)	0,4217
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 13 и более)	0,4163
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 3)	0,2691
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 4)	0,3182
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 5)	0,3617
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 6)	0,3833
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 7)	0,4039
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 8)	0,3716
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 9)	0,3885
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 10)	0,3722
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 11 и более)	0,3827

Для метода опорных векторов и случайного леса, добавление возраста позволило незначительно улучшить работу классификатора, но для дерева решений оно дало обратный эффект. Наилучший результат дал алгоритм случайного леса с 20 деревьями по энтропийному критерию с максимальной глубиной 7 - 0,4321.

Из этого можно сделать вывод, что возраст пациента коррелирует со значением целевой функции и, в целом, положительно влияет на качество модели.

3.3.2. Использование биохимического анализа крови

До этого момента для создания выборки мы использовали только данные основного клинического анализа крови и возраст. Попробуем использовать данные других анализов и обследований пациентов, которые, возможно окажутся более важными для классификатора.

В то же время это должны быть признаки, которые присутствуют у большинства или почти у всех пациентов. Узнаем какие действия наиболее распространены среди них. Для этого была собраны значения actionType_id в таблице Action для пациентов из выборки, и расшифрованы с помощью полей id, name в таблице Action (см. Рисунок 8, Таблицу 18).

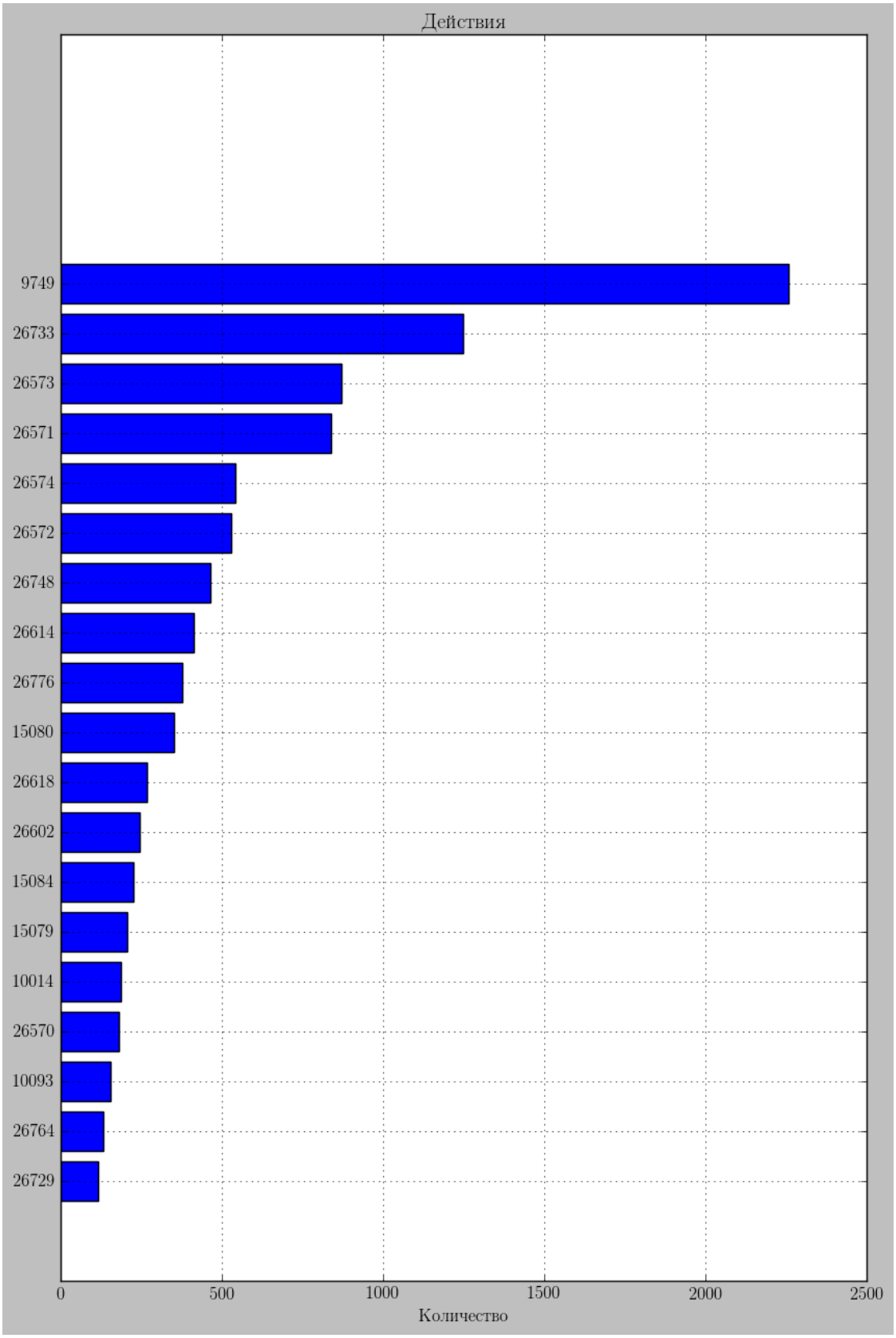


Рисунок 8. Самые распространённые действия среди пациентов.

Таблица 18. Самые распространённые действия среди пациентов.

actionType_id	Наименование
9749	Температурный лист
26733	Дневник ОАР ОНМК
26573	Биохимический анализ крови
26571	Общий клинический анализ крови
26574	Клинический анализ мочи
26572	Коагулологические исследования
26748	Базовый осмотр ОАР ОНМК
26614	Дневниковая запись врача - Невролога
26776	Осмотр ОАР ОНМК-БАЗА
15080	Движение
26618	Выписной эпикриз
26602	Дневниковая запись врача – Кардиолога
15084	Поступление
15079	Выписка
10014	Дневник ОСХ
26570	Группа крови
10093	Лекарственная терапия (препарат)
26764	Осмотр ОАР ОНМК-БАЗА
26729	Базовый осмотр 1-го неврологического отделения РСЦ

Осмотры, дневники не содержат количественных признаков, которые требуются нам. Температурный лист содержит в себе только температуру пациента, и, скорее всего, по ней сложно будет сделать вывод о болезни пациента. Поэтому, возьмем биохимический анализ крови, так как он наиболее распространён. Он дает информацию о работе внутренних органов (печени, почек, поджелудочной железы) и метаболизме [20]. Анализ включает в себя 8 числовых характеристик:

1. Креатин.
2. Общий белок.
3. Мочевина.

4. Калий.
5. АСТ (аспартатаминотрансфераза).
6. АЛТ (аланинаминотрансфераза).
7. Натрий.
8. Глюкоза.

Были составлены выборки с учетом и без учета возраста пациента, использованы метод опорных векторов, дерево решений и случайный лес и получены следующие результаты (см. Таблицу 19, Таблицу 20):

Таблица 19. Оценки классификаторов, определяющих болезнь пациента по биохимическому анализу крови.

Метод классификации	**Правильность**
SVM (с линейным ядром)	0,2749
SVM (с полиномиальным ядром)	0,3119
SVM (с RBF ядром)	0,4298
Дерево решений (по критерию Джини с максимальной глубиной 3)	0,2621
Дерево решений (по критерию Джини с максимальной глубиной 4)	0,2436
Дерево решений (по критерию Джини с максимальной глубиной 5)	0,2366
Дерево решений (по критерию Джини с максимальной глубиной 6)	0,2738
Дерево решений (по критерию Джини с максимальной глубиной 7)	0,2899
Дерево решений (по критерию Джини с максимальной глубиной 8)	0,3190
Дерево решений (по критерию Джини с максимальной глубиной 9)	0,3126
Дерево решений (по критерию Джини с максимальной глубиной 10)	0,3166
Дерево решений (по критерию Джини с максимальной глубиной 11 и более)	0,3126
Дерево решений (по энтропийному критерию с максимальной глубиной 3)	0,2196
Дерево решений (по энтропийному критерию с максимальной глубиной 4)	0,2152
Дерево решений (по энтропийному критерию с максимальной глубиной 5)	0,2633
Дерево решений (по энтропийному критерию с максимальной глубиной 6)	0,2764

Дерево решений (по энтропийному критерию с максимальной глубиной 7)	0,3003
Дерево решений (по энтропийному критерию с максимальной глубиной 8)	0,3171
Дерево решений (по энтропийному критерию с максимальной глубиной 9)	0,2988
Дерево решений (по энтропийному критерию с максимальной глубиной 10)	0,3108
Дерево решений (по энтропийному критерию с максимальной глубиной 11 и более)	0,3048
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 3)	0,2580
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 4)	0,2561
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 5)	0,3336
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 6)	0,3637
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 7)	0,3215
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 8)	0,3980
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 9)	0,4084
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 10)	0,4272
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 11 и более)	0,4278
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 3)	0,2861
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 4)	0,2824
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 5)	0,3456

Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 6)	0,3871
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 7)	0,3571
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 8)	0,3620
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 9)	0,3694
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 10)	0,3994
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 11)	0,3986
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 12 и более)	0,3871
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 3)	0,2693
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 4)	0,2430
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 5)	0,3341
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 6)	0,3221
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 7)	0,3283
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 8)	0,3708
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 9)	0,3943
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 10)	0,3885
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 11)	0,3774
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 12 и более)	0,3708

Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 3)	0,2572
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 4)	0,3048
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 5)	0,3501
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 6)	0,3637
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 7)	0,3172
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 8)	0,3561
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 9)	0,3581
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 10)	0,3345
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 11 и более)	0,3287

Таблица 20. Оценки классификаторов, определяющих болезнь пациента по биохимическому анализу крови и возрасту.

Метод классификации	**Правильность**
SVM (с линейным ядром)	0,2693
SVM (с полиномиальным ядром)	0,3069
SVM (с RBF ядром)	0,4298
Дерево решений (по критерию Джини с максимальной глубиной 3)	0,2565
Дерево решений (по критерию Джини с максимальной глубиной 4)	0,2740
Дерево решений (по критерию Джини с максимальной глубиной 5)	0,2308
Дерево решений (по критерию Джини с максимальной глубиной 6)	0,3057
Дерево решений (по критерию Джини с максимальной глубиной 7)	0,3234
Дерево решений (по критерию Джини с максимальной глубиной 8)	0,3159
Дерево решений (по критерию Джини с максимальной глубиной 9)	0,3102
Дерево решений (по критерию Джини с максимальной глубиной 10 и	0,3283

более)	
Дерево решений (по энтропийному критерию с максимальной глубиной 3)	0,2690
Дерево решений (по энтропийному критерию с максимальной глубиной 4)	0,2177
Дерево решений (по энтропийному критерию с максимальной глубиной 5)	0,2896
Дерево решений (по энтропийному критерию с максимальной глубиной 6)	0,2528
Дерево решений (по энтропийному критерию с максимальной глубиной 7)	0,2816
Дерево решений (по энтропийному критерию с максимальной глубиной 8)	0,2876
Дерево решений (по энтропийному критерию с максимальной глубиной 9)	0,2876
Дерево решений (по энтропийному критерию с максимальной глубиной 10)	0,3114
Дерево решений (по энтропийному критерию с максимальной глубиной 11 и более)	0,3173
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 3)	0,3365
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 4)	0,3877
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 5)	0,3408
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 6)	0,3928
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 7)	0,4048
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 8)	0,3698
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 9)	0,3743
Случайный лес (с 20 деревьями по критерию Джини с максимальной	0,3702

глубиной 10)	
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 11)	0,3873
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 12 и более)	0,3811
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 3)	0,3173
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 4)	0,3591
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 5)	0,3293
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 6)	0,3237
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 7)	0,3819
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 8)	0,3945
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 9)	0,3939
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 10)	0,3824
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 11 и более)	0,3879
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 3)	0,2954
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 4)	0,2993
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 5)	0,3287
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 6)	0,3706
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 7)	0,3737
Случайный лес (с 10 деревьями по критерию Джини с максимальной	0,3523

глубиной 8)	
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 9)	0,3507
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 10)	0,3569
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 11)	0,3509
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 12 и более)	0,3450
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 3)	0,3132
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 4)	0,2997
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 5)	0,3116
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 6)	0,3137
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 7)	0,3172
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 8)	0,3606
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 9)	0,3600
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 10)	0,3772
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 11 и более)	0,3661

Замена признаков на биохимический анализ крови не помогла улучшить качество модели и даже ухудшила ее. Значения метрик качества получились ниже, чем при использовании основного клинического анализа крови. Включение возраста в список признаков немного повысило правильность модели, но в целом, использование числовых характеристик биохимического

анализа крови при обучении дает более низкие показатели, чем при использовании основного клинического анализа крови.

Глава 4. Решение проблемы несбалансированных выборок

В обеих задачах – и по диагностированию операций, и по определению болезни пациента присутствуют несбалансированные данные, т. е. объектов одного или нескольких классов значительно меньше или больше, чем других. При этом класс с большим количеством элементов называется доминирующим, а с маленьким – минорным.

Если во второй задаче, разница в размере классов не так велика, то в первой один из классов составляет менее 5% другого (положительный класс – 3 элемента, отрицательный класс – 76 элементов). В такой ситуации рекомендуется собрать больше данных, но, к сожалению, у нас нет сейчас такой возможности [21].

В этом случае рекомендуется использовать методы изменения размера выборок – увеличение размера минорного класса (oversampling) и уменьшение размера доминирующего класса (undersampling). Так как полученные выборки имеют небольшой размер, то было бы рационально использовать для обеих выборок oversampling, но найденные алгоритмы увеличения размера выборки предназначены только для бинарной классификации.

Поэтому, для задачи определения болезни будет использоваться undersampling.

4.1. Oversampling

4.1.2. Алгоритм SMOTE

Алгоритм SMOTE (Synthetic Minority Over-sampling Technique) был создан в 2002 году Н. В. Чавлой, К. У. Бойером, Л. О. Холлом и В. П. Кегельмейером. Он заключается в искусственном генерировании дополнительных элементов для минорного класса, используя данные об уже существующих.

Последовательность действий алгоритма следующая:

1. Берем случайный элемент из минорного класса x_1, где x_1 – вектор, содержащий в себе числовые характеристики признаков объекта;
2. Из k ближайших соседей берем случайный элемент x_2 из этого же минорного класса (k является настраиваемым параметром для алгоритма);
3. Находим новый элемент миноритарного класса следующим образом: $\alpha(x_1 - x_2) + x_2$, α – случайное число из промежутка (0;1).

Генерация новых элементов происходит, пока это необходимо (количество генерируемых элементов также является настраиваемым параметром) [22].

4.1.2. Алгоритм ADASYN

Алгоритм ADASYN (adaptive synthetic) был создан в 2008 году Хайбо Хэ, Янг Бэй, Э. А. Гарсией и Шутао Ли. По сути, он является модифицированной версией алгоритма SMOTE.

Пусть у нас есть тренировочная выборка D_{tr}, в которой содержится m объектов типа $\{x_i, y_i\}, i = \overline{1, m}$, где x_i – вектор с числовыми характеристиками объекта, а y_i – значение целевой функции для него. Обозначим m_s и m_l как количество элементов в минорном и доминирующем классе соответственно.

1. Посчитаем $d = \frac{m_s}{m_l}$ – степень дисбаланса классов (если $d = 1$, то классы одинакового размера).
2. Если $d < d_{th}$, где d_{th} - заданный порог для максимально допустимой степени классового дисбаланса:
    a. Найдем количество элементов, которые необходимо сгенерировать для минорного класса: $G = (m_l - m_s) \times \beta$, где β – настраиваемый параметр для регулирования количества генерируемых элементов минорного класса (при β=1 количество элементов в минорном классе станет равным количеству элементов в доминирующем);

b. Для каждого x_i, принадлежащего минорному классу, найдем k ближайших соседей и посчитаем $r_i = \frac{\Delta_i}{k}$, где Δ_i – количество соседей, которые принадлежат доминирующему классу;
c. Проведем нормировку величины r_i: $\widehat{r_i} = \frac{r_i}{\sum_{i=1}^{m_s} r_i}$
d. Подсчитаем количество генерируемых примеров g_i для каждого элемента минорного класса x_i: $g_i = \widehat{r_i} \times G$
e. Для каждого элемента минорного класса x_i найдем g_i новых элементов. Для каждого x_i g_i раз выполняем следующую процедуру:
    i. Случайно выбираем из k ближайших соседей элемента x_i элемент минорного класса x_{zi}
    ii. Генерируем новый элемент минорной выборки $s_i = (x_{zi} - x_i) \times \alpha + x_i$, где α – случайное число из промежутка [0;1].

Ключевой идеей алгоритма является использование весов $\widehat{r_i}$, которые определяют количество генерируемых примеров для каждого элемента минорной выборки. Впоследствии, он не только обеспечит сбалансированность выборок, но и поможет алгоритму обучению на трудных для обучения примерах. В этом и заключается его главное отличие от алгоритма SMOTE [23].

4.2. Undersampling

В данной работе используется простейший алгоритм уменьшения размера доминирующего класса – Random Undersampling. Он случайным образом выбирает и удаляет элементы из доминирующего класса, пока не будет достигнуто желаемое количество элементов в классе [24]. Плюсом этого метода является то, что он также подходит для многоклассовой классификации.

4.3. Результаты для классификатора по операциям

Правильным подходом было бы использовать алгоритмы увеличения размера минорного класса только на тестовой выборке. Но так как в нашем случае в минорном классе всего 3 элемента, было принято решение применить oversampling ко всей выборке, иначе при делении выборки в отношении 2:1 ($\frac{2}{3}$ – тренировочное множество и $\frac{1}{3}$ – тестовое множество) в тестовом множестве оказался бы всего лишь 1 экземпляр минорного класса. Этих данных было бы недостаточно, чтобы объективно оценить работу классификатора.

Для применения алгоритмов SMOTE и ADASYN был применен язык программирования Python версии 3.5 и библиотека для работы с несбалансированными выборками Imbalanced-learn [25]. Для алгоритма SMOTE параметр k, характеризующий число ближайших соседей, был взят равным 2, так как на начальном этапе размер минорной выборки равен 3. Для алгоритма ADASYN использовалась значение k, равное 5 (значение по умолчанию, $\beta = 1$. После того, как размеры минорной и доминирующей выборки стали равны, были применены метод опорных векторов, дерево решений и случайный лес с использованием кросс-валидации, были получены следующие результаты (см. Таблицу 21, Таблицу 22):

Таблица 21. Оценки классификаторов, принимающих решение о необходимости операции после применения алгоритма SMOTE.

Метод классификации	**precison**	**recall**	**f-мера**	**accuracy**
SVM (с линейным ядром)	0,9404	1,0	0,9692	0,9681
SVM (с полиномиальным ядром)	0,9251	1,0	0,9592	0,9551
SVM (с RBF ядром)	1,0	0.4677	0,6324	0.7338
Дерево решений (по критерию Джини с максимальной глубиной 3)	0,9182	0,9753	0,9447	0,9430
Дерево решений (по критерию Джини с максимальной глубиной 4 и более)	0,9288	0,9753	0,9506	0,9494
Дерево решений (по энтропийному критерию с максимальной глубиной 3)	0,9182	0,9753	0,9447	0,9430

Дерево решений (по энтропийному критерию с максимальной глубиной 4 и более)	0,9288	0,9753	0,9506	0,9494
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 3)	0,8865	0,9876	0,9341	0,9304
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 4 и более)	0,9178	0,9876	0,9514	0,9496
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 3)	0,8853	0,9753	0,9277	0,9242
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 4)	0,8960	0,9753	0,9335	0,9306
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 5)	0,9170	0,9753	0,9447	0,9432
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 6 и более)	0,9276	0,9753	0,9506	0,9496
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 3)	0,8853	0,9753	0,9277	0,9242
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 4)	0,9059	0,9753	0,9391	0,9370
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 5 и более)	0,9170	0,9753	0,9447	0,9432
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 3)	0,8760	0,9753	0,9222	0,9178
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 4)	0,8960	0,9753	0,9335	0,9306
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 5)	0,9170	0,9753	0,9447	0,9432
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 6 и более)	0,9160	0,9629	0,9380	0,9370

Таблица 22. Оценки классификаторов, принимающих решение о необходимости операции после применения алгоритма ADASYN.

Метод классификации	**precison**	**recall**	**f-мера**	**accuracy**
SVM (с линейным ядром)	0,9220	0,7901	0,8174	0,8632
SVM (с полиномиальным ядром)	0,9191	0,9615	0,9393	0,9361

SVM (с RBF ядром)	1,0	0,2749	0,3932	0,6374
Дерево решений (по критерию Джини с максимальной глубиной 3)	0,8732	0,7777	0,7890	0,8380
Дерево решений (по критерию Джини с максимальной глубиной 4 и более)	0,8949	0,7777	0,8007	0,8509
Дерево решений (по энтропийному критерию с максимальной глубиной 3)	0,8732	0,7777	0,7890	0,8380
Дерево решений (по энтропийному критерию с максимальной глубиной 4 и более)	0,8949	0,7777	0,8007	0,8509
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 3)	0,9404	0,7516	0,7974	0,8437
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 4)	0,9642	0,8390	0,8800	0,9002
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 5 и более)	0,9540	0,8143	0,8530	0,8815
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 3)	0,9281	0,7763	0,8161	0,8497
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 4)	0,9506	0,8133	0,8607	0,8810
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 5 и более)	0,9506	0,7886	0,8393	0,8687
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 3)	0,9135	0,6115	0,6915	0,7737
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 4)	0,9540	0,8390	0,8744	0,8938
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 5 и более)	0,9540	0,8019	0,8414	0,8753
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 3)	0,9104	0,7640	0,8043	0,8437
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 4)	0,8437	0,8133	0,8607	0,8810
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 5 и более)	0,9506	0,7763	0,8277	0,8625

Как можно наблюдать, использование алгоритмов SMOTE и ADASYN

привело практически к переобучению всех классификаторов, кроме метода опорных векторов с RBF ядром. В обоих случаях он не обнаружил все операции, зато не поставил не одной ошибочной (т. к. $precision = 1$), что в принципе является неплохим результатом. Однако, все равно неизвестно, как он будет работать на реальных данных, т. к. обучающая выборка была частично искусственно сгенерирована.

4.4. Результаты для классификатора, определяющего болезнь

Была взята выборка, содержащая сведения об основном клиническом анализе крови и возрасте пациента, т. к. до этого она показала наилучшие результаты. Выборка была поделена в отношении 2:1 на тренировочное и тестовое множество соотвественно. К тренировочному множеству был применен алгоритм Random Undersampling, размер для всех классов в тренировочной выборке стала одинаковым. При применении метода опорных векторов, дерева решений и случайного леса были получены следующие результаты (см. Таблицу 22):

Таблица 22. Оценки классификаторов, определяющих болезнь пациента по основному клиническому анализу крови и возрасту при использовании Random Undersampling.

Метод классификации	**Правильность**
SVM (с линейным ядром)	0,1290
SVM (с полиномиальным ядром)	0,2419
SVM (с RBF ядром)	0,4516
Дерево решений (по критерию Джини с максимальной глубиной 3)	0,3387
Дерево решений (по критерию Джини с максимальной глубиной 4)	0,2903
Дерево решений (по критерию Джини с максимальной глубиной 5)	0,3064
Дерево решений (по критерию Джини с максимальной глубиной 6)	0,2903
Дерево решений (по критерию Джини с максимальной глубиной 7 и более)	0,3064

Дерево решений (по энтропийному критерию с максимальной глубиной 3)	0,3709
Дерево решений (по энтропийному критерию с максимальной глубиной 4)	0,2903
Дерево решений (по энтропийному критерию с максимальной глубиной 5)	0,3225
Дерево решений (по энтропийному критерию с максимальной глубиной 6 и более)	0,2741
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 3)	0,2903
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 4)	0,3709
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 5)	0,3870
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 6)	0,3709
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 7)	0,3548
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 8)	0,3387
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 9)	0,4032
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 10)	0,3870
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 11)	0,4193
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 12)	0,4354
Случайный лес (с 20 деревьями по критерию Джини с максимальной глубиной 13 и более)	0,4193
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 3)	0,3387
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 4)	0,3387

Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 5)	0,4516
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 6)	0,4032
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 7)	0,4032
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 8)	0,4516
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 9)	0,4193
Случайный лес (с 20 деревьями по энтропийному критерию с максимальной глубиной 10 и более)	0,4354
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 3)	0,3387
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 4)	0,3548
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 5)	0,4193
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 6)	0,3870
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 7)	0,4193
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 8)	0,3548
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 9)	0,3870
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 10)	0,3709
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 11)	0,3870
Случайный лес (с 10 деревьями по критерию Джини с максимальной глубиной 12 и более)	0,4032
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 3)	0,2580

Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 4)	0,3548
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 5)	0,4032
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 6)	0,4354
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 7)	0,3709
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 8)	0,3709
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 9)	0,4032
Случайный лес (с 10 деревьями по энтропийному критерию с максимальной глубиной 10 и более)	0,4193

В целом, качество полученных моделей ухудшилось по сравнению с аналогичными результатами в главе 2. Некоторые методы при определенных параметрах показали улучшение, но незначительное. Учитывая малый размер выборки пациентов, лучше метод Random Undersampling не использовать.

Выводы

В работе была рассмотрена современная МИС «Виста-Мед». Изучен ее интерфейс, устройство и архитектура баз данных.

Рассмотрены методы машинного обучения и метрики качества, позволяющие оценить их работу: точность (precision), полнота (recall), f-мера, правильность (accuracy).

Из SQL-дампа были извлечены данные о пациентах. Было найдено самое распространённое заболевание (инфаркт мозга) и собрана необходимая информация, позволяющая сделать вывод о необходимости операции – основной клинический анализ крови. Полученная выборка оказалась несбалансированной, т.е. размер одного класса значительно меньше, чем другого (3 пациента с операцией и 79 без операции).

Рассмотрены следующие алгоритмы: метод опорных векторов, дерево решений, случайный лес. Для использования методов машинного обучения была изучена специализированная библиотека Scikit-learn. Результаты использования на несбалансированной выборке оставляли желать лучшего, т. к. классификатор не мог обучиться на таком маленьком объеме данных.

Задача была обобщена до классификации пациентов по болезням. Необходимые данные были собраны из дампа SQL-базы, полученная выборка также оказалась несбалансированной. Были испробованы различные численные характеристики признаков пациентов: основной клинический анализ крови, биохимический анализ крови. Наилучший результат дал основной клинический анализ крови. Был добавлен возраст пациента, что помогло увеличить значения мер качества, и, как следствие, получить более качественную модель машинного обучения.

Были рассмотрены варианты решения проблем несбалансированных выборок – уменьшение размера доминирующего класса (undersampling) и увеличение размера минорного класса (oversampling). Изучена библиотека для работы с несбалансированными выборками Imbalanced-learn. Были применены

алгоритмы SMOTE, ADASYN для задачи о принятии решения об операции и алгоритм Random Undersampling для задачи классификации пациентов по болезням.

Заключение

Были разработаны следующие прототипа информационных систем:

1. Экспертная система, позволяющая принять решение о необходимости операции тому или иному пациенту с инфарктом мозга. В дальнейшем, планируется собрать больше данных, и улучшить качество его работы.

2. Экспертная система, позволяющая классифицировать пациентов по болезням.

Для создания систем, использовались следующие методы: метод опорных векторов, дерево решений и случайный лес. Также, для увеличения размера минорной выборки (в пункте 1) были использованы алгоритмы SMOTE, ADASYN.

В дальнейшем, планируется собрать больше данных, получить экспертные системы должного качества и при положительном результате встроить полученные продукты в МИС «Виста-мед» в качестве рекомендательной системы для врача.

В последнее время также получают распространение телемедицинские технологии. Они позволяют высококвалифицированным врачам удаленно участвовать в лечении пациентов, тем самым повышая его качество.

Список литературы

1. Souillard-Mandar, W., Davis, R., Rudin, C. Learning classification models of cognitive conditions from subtle behaviors in the digital Clock Drawing Test // Machine Learning, 2016. Vol. 102, Issue 3, P. 393–441.
2. ERP definition - Enterprise Resource Planning - Gartner IT. http://www.gartner.com/it-glossary/enterprise-resource-planning-erp/
3. Гусев А. В. Рынок медицинских информационных систем: обзор, изменения, тренды // Врач и информационные технологии, 2012. №3, С. 6-15.
4. G. E. A. P. A. Batista, R. C. Prati, M. C. Monard. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data // ACM SIGKDD Explorations Newsletter - Special issue on learning from imbalanced datasets, 2004. Vol. 6, Issue 1, P.20-29.
5. MySQL Workbench Manual https://dev.mysql.com/doc/workbench/en/
6. Воронцов К. В. Математические методы обучения по прецедентам (теория обучения машин) http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf
7. Всемирная организация здравоохранения. Международная статистическая классификация болезней и проблем, связанных со здоровьем; 10-й пересмотр. Том 1. М.: Медицина, 1995. 698 с.
8. Ишемический инсульт – Заболевания. Национальный медико-хирургический Центр имени Н. И. Пирогова. http://old.pirogov-center.ru/illness/24/
9. MySQLdb User's Guide http://mysql-python.sourceforge.net/MySQLdb.html
10. Matplotlib 2.0.1 documentation https://matplotlib.org/
11. Линейные методы классификации и регрессии: метод опорных векторов http://www.machinelearning.ru/wiki/images/a/a0/Voron-ML-Lin-SVM.pdf
12. Оре О. Теория графов. 2-е изд. М.: Наука, 1980. 336 с.

13. Воронцов К. В. Лекции по логическим алгоритмам классификации http://www.machinelearning.ru/wiki/images/3/3e/Voron-ML-Logic.pdf
14. L. Breiman, J.H. Friedman, R.A. Olshen, and C.T. Stone. Classification and Regression Trees. Wadsworth, Belmont, California, 1984.
15. Дерево принятия решений http://www.amse.ru/archive/courses/2006/nikolenko/notes-01-dectrees.pdf
16. L. Breiman. Random forests // Machine Learning, 2001. Vol. 45, Issue 1, P. 5–32.
17. Маннинг К., Рагхаван П., Шютце Х. Введение в информационный поиск. М.: Вильямс, 2011. 528 с.
18. Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection // In Proceedings of the 14th international joint conference on Artificial intelligence, 1995. - Volume 2 (IJCAI'95), Vol. 2. P. 1137-1143.
19. Documentation scikit-learn: machine learning in Python http://scikit-learn.org/stable/documentation.html
20. Биохимический анализ крови - Диагностика заболеваний http://medportal.ru/enc/analysis/diseases/14/
21. 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset - Machine Learning Mastery http://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/
22. Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, W. Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique // Journal of Artificial Intelligence Research, 2002. Vol. 16, Issue 1. P. 321-357
23. Haibo He, Yang Bai, Edwardo A. Garcia, Shutao Li. ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning // International Joint Conference on Neural Networks (IJCNN), 2008. P. 1322-1328
24. J. Prusa, T. M. Khoshgoftaar, D. J. Dittman and A. Napolitano. Using Random Undersampling to Alleviate Class Imbalance on Tweet Sentiment Data // IEEE International Conference on Information Reuse and Integration, 2015.

P. 197-202.

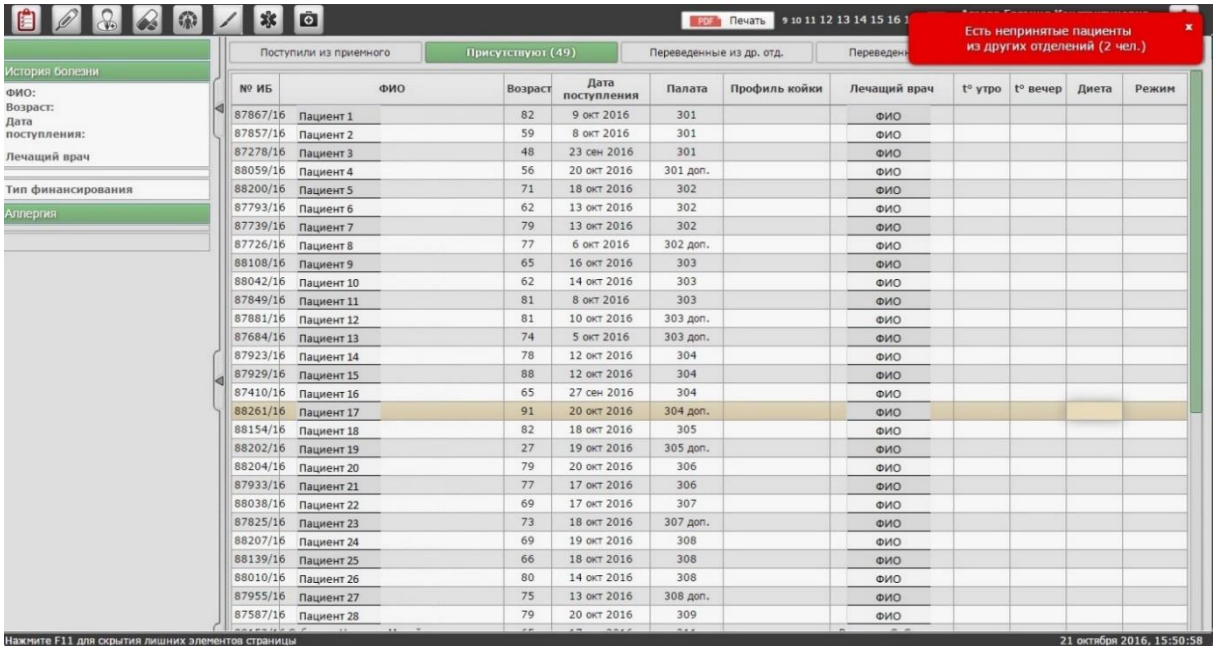
25. Imbalanced-learn 0.3.0.dev0 documentation http://contrib.scikit-learn.org/imbalanced-learn/index.html

Приложение

Приложение 1

На рисунке 1 изображен интерфейс записи пациентов в палаты. Доступ к нему осуществляется через профиль медсестры.

Рисунок 1. Распределение пациентов по палатам.



Также имеется возможность просматривать список выполненных в учреждении диагностических исследований. Это позволяет отслеживать какие исследования проводятся пациентам. Пример на рисунке 2.

Рисунок 2. Список диагностических исследований.